

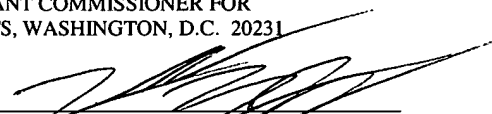
PATENT
5053-31001

"EXPRESS MAIL" MAILING LABEL NUMBER

EL675026073US

DATE OF DEPOSIT: OCTOBER 27, 2000

I HEREBY CERTIFY THAT THIS PAPER OR
FEE IS BEING DEPOSITED WITH THE
UNITED STATES POSTAL SERVICE
"EXPRESS MAIL POST OFFICE TO
ADDRESSEE" SERVICE UNDER 37 C.F.R. §
1.10 ON THE DATE INDICATED ABOVE
AND IS ADDRESSED TO BOX
PROVISIONAL PATENT APPLICATION,
ASSISTANT COMMISSIONER FOR
PATENTS, WASHINGTON, D.C. 20231


Derrick Brown

**SMART TRIGGER FOR USE
IN PROCESSING BUSINESS TRANSACTIONS**

By:

Robert Jay Shaw

Citizenship: USA
4312 Seabury
Dallas, Texas 78287

Attorney Docket No.: 5053-31001

Eric B. Meyertons/RPH
Conley, Rose & Tayon, P.C.
P.O. Box 398
Austin, Texas 78767-0398
Ph: (512) 476-1400

PRIORITY CLAIM

This application claims the benefit of U.S. Provisional Application No. 60/162,602 entitled "Smart Trigger For Use In Processing Business Transactions," filed
5 October 29, 1999.

BACKGROUND OF THE INVENTION

1. Field of the Invention

10

The present invention generally relates to computer software programs and databases to be used in Financial Service Organizations (FSO's). More particularly, the present invention relates to the selective identification and execution of a specific processing task for one or more records contained in the one or more Financial Service
15 Organization (FSO) data sets.

2. Description of the Related Art

FSOs such as banks, credit unions, etc., use computer systems running software
20 programs to process customer account transactions. The computer systems may include databases for storing data such as the master files of customer account information, transactional data such as customer credit card purchase transactions, processing data such as the processing parameters used in processing transactions, and history data such as log files of daily activities for later batch processing.

25

Databases may be used in FSO business transaction processing systems to store, manage and retrieve data for a variety of applications. In many instances, the databases may be extremely large. The contents of the database may often occupy memory space measured in hundred's of gigabytes. When application programs need to access

enormous amounts of data on a transactional basis, whether in batch mode or real-time mode, the FSO business transaction processing systems often utilize commercially available database products. One embodiment of a commercially available database product is the DB2 database from International Business Machines (IBM).

5

Some nomenclature is introduced here to aid in the understanding of terminology used within the FSO business transaction processing system. A trigger is a defined set of actions that are executed when a delete, insert, or update operation is carried out against a specified table in the FSO database. Stored procedures are similar to a trigger. Both
10 consist of procedural logic that is stored at the database level. However, stored procedures are not event-driven and are not attached to a specific table. A stored procedure is explicitly executed by invoking a CALL to the procedure (instead of implicitly being executed like triggers). Additionally, a stored procedure can access many tables without being specifically associated to any of them.

15

An example of an FSO that may use such a business transaction processing system is a credit card institution. A credit card institution may issue credit cards to customers of the FSO. The credit card institution may also issue credit cards on behalf of client businesses such as department stores. The credit card institution may also acquire
20 and process credit card transactions from customers and client businesses such as department stores. For example, a credit card institution may issue its own credit card. The credit card institution may also have a client department store. The credit card institution may issue a credit card under the department store's name, and may collect and process all credit card transactions for the department store, charging a fee for each
25 transaction processed. Some of the credit card transactions collected by the credit card institution may be for credit cards not issued by the credit card institution. These transactions may be forwarded to the FSO that issued the card. In turn, other FSOs may forward credit card transactions to the credit card institution. Transactions for credit cards issued by the credit card institution may be processed by the credit card institution.

The FSO business transaction processing system may include a data dictionary. A data dictionary may be defined as a collection of descriptions of the data items or elements in the database. For example, the FSO business transaction processing system data dictionary may describe the data elements involved in credit card processing. The data dictionary may describe each of the data elements in the database for credit card processing. Data sets are groups of data, such as master files and transactions, and may be comprised of data elements defined in the data dictionary. Examples of data elements in the FSO data dictionary are customer name, credit card type, and card issuer.

10

The FSO business transaction processing system may include processing parameters used in processing transactions. Processing parameters may be used to apply business logic to the data elements in the transaction during processing. An example of a transaction in the FSO system is a credit card transaction. An example of a processing parameter is a transaction price that may be charged to a client of a credit card institution for processing a credit card transaction.

The FSO transaction processing application software program may use one or more processing parameters while processing a transaction. A processing parameter may have different values for different transactions. The application software program may examine the values of one or more data elements in the transaction data or database master files to determine the value of a processing parameter for the transaction.

The FSO system database may include processing parameters used in processing transactions. Processing parameters may be used to apply business logic to the data elements in the transaction during processing. An example of a transaction in an FSO system is a credit card transaction. An example of a processing parameter is a transaction price that may be charged to a client of a credit card institution for processing a credit card transaction.

An FSO transaction processing software program may use one or more processing parameters while processing a transaction. A processing parameter may have different values for different transactions. The software program may examine the values of one or more data elements in the transaction data or master files to determine the value of a processing parameter for the transaction.

In one embodiment, an FSO transaction processing system may process one or more data sets using schedule based processing. The user of the FSO may schedule a period for execution of a processing task for the data set. On meeting the qualifications of the schedule, the scheduled processing tasks would be automatically executed.

In another embodiment, an FSO transaction processing system may process data sets using random event based processing. The user of the FSO may at random or based on an event request the execution of a processing task for the data set. On meeting the qualifications, the requested processing task would be automatically executed.

In prior art and in one embodiment, the processing of data sets or master files was periodic and 'account centric'. Application programs in the FSO computer system had to be scheduled to read and evaluate sequentially every record in the data set to determine if one or more processing tasks were to be executed for the record. For example, a data set may contain a list of all account numbers for an FSO. Processing of an account at a pre-determined period, may comprise evaluating conditions to execute for each account number a list of one or more functions or processing tasks. Preferred embodiments of the processing tasks performed on an account number may require the FSO to raise credit limit, change expiration date, send card data for embossing, and others. For example, the FSO may need to extend the expiration date on a group of credit card accounts, which have a current expiration date of 12/99. The application program would sequentially access every record in an account master file and determine which one of the one or more

functions or processing tasks is needed to be performed. Each record lacks information that identifies the particular processing task to be performed. This requires that each of the one or more functions or processing tasks must be queried to determine if it is applicable. If master file record matches the specified criteria of having an expiration date of 12/99, then it would execute the processing task that is subsequently found to be applicable i.e. extend the date of expiration of the credit card. If the evaluation criteria did not match then it would go to the next record in the FSO account master file. Assuming a master file has millions of records, and each record may have one or more processing tasks associated with it, the process of sequentially accessing every record to determine applicability of a processing task to a specific record is very inefficient and time consuming. The method also utilizes extensive FSO computer resources to process FSO data sets.

A conventional method to process FSO data sets in a random event based manner, may use a trigger function provided by IBM's DB2 database sub-system. This method limits FSO data set processing to a specified table. Processing of FSO data sets may need a trigger function to access multiple DB2 tables. Another embodiment of prior art, which is also based on IBM's DB2 database sub-system, may use a stored procedure to process FSO data sets. This method has one or more drawbacks, which may limit the usefulness in FSO data set processing applications. Stored procedures are not event driven and must be explicitly initiated by a CALL command in an application program of the FSO computer system.

It is, therefore, desirable to provide a method and a system to identify and execute only those processing tasks for an FSO data set which have been identified to need further processing. The Smart Trigger method and system to process FSO data sets, improves on the trigger and stored procedure methods by removing their drawbacks and uses periodic and event based techniques. The improved method and system is thus more 'function centric' and not 'account centric'.

30

SUMMARY OF THE INVENTION

The present invention provides various embodiments of an improved method and system for the selective identification and execution of a specific processing task for one or more records contained in the one or more Financial Service Organization (FSO) data sets. An FSO computer user may configure a smart trigger table by defining a scheduled date, an identifier for a data set and an associated processing task. In one embodiment, an identifier for a data set may be a pointer to database memory location of the data set. In one embodiment, an activity number and its corresponding activity data may define an associated processing task. In one embodiment, the specified scheduled date may determine the trigger condition.

In one embodiment, the FSO user may schedule a task to be executed at a specified interval to process entries in the smart trigger table. In one embodiment, the task may be defined to run on a periodic basis. If conditions have been met to process the smart trigger table then each row in the smart trigger table may be processed. In one embodiment, the current date and time may be used to compare to the specified interval of the scheduled task to determine if the conditions to initiate the processing of entries in the smart trigger table have been met. If the scheduled conditions have not been met then the scheduled task may wait. If the scheduled conditions have been met i.e., the specified interval has expired, then the scheduled task may initiate processing of each row in the smart trigger table. In one embodiment, only those rows of the smart trigger table, which have met the trigger conditions, may be processed by executing the associated processing task specified in the smart trigger table record. In one embodiment, the trigger condition may be met if the scheduled date is less than or equal to current date.

After processing the smart trigger table record, i.e., executing the associated processing task specified in the smart trigger table record, in one embodiment, the

scheduled date entry for the processed smart trigger table record may be reset to indicate its completion.

BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 is a prior art block diagram illustrating one embodiment of a financial service organization business transaction processing system;

5 Figure 2 illustrates one embodiment of a financial service organization business transaction processing system with Smart Trigger software;

Figure 3a is a data flow diagram illustrating a Smart Trigger selection for an associated executable processing task according to one embodiment;

10 Figure 3b is a diagram illustrating a Smart Trigger date dependent data set selection according to one embodiment;

Figure 4a is a flowchart illustrating the configuration and runtime use of prior art data set processing method for an FSO computer system, according to one embodiment;

Figure 4b is a continuation of the flowchart in Figure 4a;

Figure 4c is a continuation of the flowchart in Figure 4b;

15 Figure 5 is a flowchart illustrating the configuration and runtime use of a Smart Trigger based processing method for an FSO computer system, according to one embodiment;

Figure 6 is a data flow diagram illustrating sequential processing of a FSO data set in prior art, according to one embodiment;

20 Figure 7 is a data flow diagram illustrating selective task processing of a FSO data set using Smart Trigger, according to one embodiment; and

Figure 8 is a block diagram illustrating a scheduled task processing entries in a Smart Trigger table, according to one embodiment.

25 While the invention is susceptible to various modifications and alternative forms, specific embodiments thereof are shown by way of example in the drawings and will herein be described in detail. It should be understood, however, that the drawings and detailed description thereto are not intended to limit the invention to the particular form disclosed, but on the contrary, the intention is to cover all modifications, equivalents, and

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

The term "computer system" as used herein generally describes the hardware and software components that in combination allow the execution of computer programs.

5 The computer programs may be implemented in software, hardware, or a combination of software and hardware. A computer system's hardware generally includes a processor, memory media, and Input/Output (I/O) devices. As used herein, the term "processor" generally describes the logic circuitry that responds to and processes the basic instructions that operate a computer system. The term "memory medium" includes an installation
10 medium, e.g., a CD-ROM, or floppy disks; a volatile computer system memory such as DRAM, SRAM, EDO RAM, Rambus RAM, etc.; or a non-volatile memory such as optical storage or a magnetic medium, e.g., a hard drive. The term "memory" is used synonymously with "memory medium" herein. The memory medium may comprise other types of memory or combinations thereof. In addition, the memory medium may be located in a first
15 computer in which the programs are executed, or may be located in a second computer that connects to the first computer over a network. In the latter instance, the second computer provides the program instructions to the first computer for execution. In addition, the computer system may take various forms, including a personal computer system, mainframe computer system, workstation, network appliance, Internet appliance,
20 personal digital assistant (PDA), television system or other device. In general, the term "computer system" can be broadly defined to encompass any device having a processor that executes instructions from a memory medium.

The memory medium preferably stores a software program or programs for
25 business modeler for defining and storing a model of an FSO production system as described herein. The software program(s) may be implemented in any of various ways, including procedure-based techniques, component-based techniques, and/or object-oriented techniques, among others. For example, the software program may be implemented using ActiveX controls, C++ objects, JavaBeans, Microsoft Foundation

Classes (MFC), or other technologies or methodologies, as desired. A CPU, such as the host CPU, executing code and data from the memory medium includes a means for creating and executing the software program or programs according to the methods, flowcharts, and/or block diagrams described below.

5

A computer system's software generally includes at least one operating system such as Windows NT available from Microsoft Corporation, a specialized software program that manages and provides services to other software programs on the computer system.

Software may also include one or more programs to perform various tasks on the computer system and various forms of data to be used by the operating system or other programs on the computer system. The data may include but are not limited to databases, text files, and graphics files. A computer system's software generally is stored in non-volatile memory or on an installation medium. A program may be copied into a volatile memory when running on the computer system. Data may be read into volatile memory as the data is required by a program.

15

A server program may be defined as a computer program that, when executed, provides services to other computer programs executing in the same or other computer systems. The computer system on which a server program is executing may be referred to as a server, though it may contain a number of server and client programs. In the client/server model, a server program awaits and fulfills requests from client programs in the same or other computer systems. An example of a computer program that may serve as a server is Windows NT server, available from Microsoft Corporation.

20

As used herein, a Financial Service Organization (FSO) is a business organization that provides financial services to customers and client organizations. As used herein, the term customer generally refers to an individual, and client organization generally refers to other businesses, including retail businesses and other FSOs. Services provided to customers and client organizations include credit products, such as loans and credit cards.

25

An FSO may also provide services to client organizations such as credit card transaction processing. Examples of FSOs include, but are not limited to, banks and credit unions. An FSO that issues credit cards and processes credit card transactions may be referred to as a credit card institution. An FSO may include one or more organizational units.

- 5 Examples of organizational units include, but are not limited to, main offices, divisions, regional offices, and branch offices.

As used herein, an FSO transaction may be defined as an occurrence of a service provided to a customer or client organization. Examples of FSO transactions include, but
10 are not limited to, financial transactions such as deposits, withdrawals, loan application servicing, and credit card application servicing. FSO transactions may also include services related to financial products such as loans and credit cards previously issued to FSO customers and client organizations. These services may include processing of credit card purchases and collection of payments.

15

An FSO system may include a data dictionary. A data dictionary may be defined as a collection of descriptions of the data items or elements in the database. For example, an FSO system data dictionary may describe the data elements involved in credit card processing. The data dictionary may describe each of the data elements in the database
20 for credit card processing. Groups of data such as master files and transaction data sets may comprise data elements defined in the data dictionary. Examples of data elements in an FSO data dictionary include, but are not limited to: customer name, credit card type, and card issuer.

25 The FSO system database may include processing parameters used in processing transactions. Processing parameters may be used to apply business logic to the transactions during processing. An example of a transaction processed in an FSO system is a credit card purchase transaction. An example of a processing parameter is a credit

card purchase transaction price that may be charged to a client of a credit card institution for the processing of a credit card purchase transaction.

5 An FSO transaction processing software program may use one or more processing parameters during the processing of a transaction. A processing parameter may have different values for different transactions. The software program may examine the values of one or more data elements in the transaction data and master files to determine the value of a processing parameter for the transaction. A combination of data elements used to determine the value of a processing parameter may be referred to as a key definition
10 for the processing parameter. The combination of data element values constructed from the key definition may be referred to as a key value. For example, a software program for processing credit card transactions for a credit card institution may use the credit card issuer and card type to determine what transaction price to charge a client of the credit card institution for processing a credit card transaction. The key definition in this
15 example includes the credit card issuer data element and card type data element, and the key value is constructed from the values for the credit card issuer data element and card type data element read from the credit card transaction data or from a master file associated with the transaction.

20 In one embodiment, processing parameters and the key values used to identify them may be stored in tables in the database. The tables in the database that store the processing parameters and keys may be referred to as Process Control Data (PCD) tables. In one embodiment, there may be one PCD table for each processing parameter in the FSO system.

25

Processing parameters are one example of parameters that may be stored in PCD tables and located using key definitions as described herein. Examples of other types of parameters that may be stored in PCD tables are default parameters and definition parameters. Default parameters may be used to fill in default information in records in

the database when they are created. For example, when a new customer account is created, one or more fields in the customer account master file may be filled with default parameter values. Default parameter values may be retrieved from PCD tables using key values constructed from the PCD key definitions and data element values from the customer account master file. Definition parameters are text or numeric values that are located using key values as codes. An example is a text error message that may be looked up using a numeric error code as a key value.

During processing, an FSO transaction may be stored as a record or file in the FSO system. In one embodiment, the FSO transaction may be stored in the FSO system database. A portion of the FSO transaction record may be read into system memory during processing. An FSO transaction record may include one or more data elements. The data elements included in an FSO transaction record may be defined in the data dictionary. The data elements in the transaction record may describe the various attributes of the transaction. For example, the data elements in a credit card transaction record may include items such as the customer's name, account numbers, credit card type, card issuer, date of the transaction, and the business at which the transaction originated.

An example of an FSO that may use an FSO computer system as described herein is a credit card institution. A credit card institution may issue credit cards to customers and client institutions of the FSO. The credit card institution may also issue credit cards on behalf of client businesses such as department stores. The credit card institution may also acquire and process credit card transactions from customers and client businesses such as department stores. For example, a credit card institution may issue its own credit card. Continuing the example, the credit card institution may also have client department stores. The credit card institution may issue a credit card under a department store's name, and may collect and process all credit card transactions for the department store. The credit card institution may charge a fee for each transaction processed. Some of the

credit card transactions collected by the credit card institution may be transactions for credit cards not issued by the credit card institution. These transactions may be forwarded to the FSO that issued the card. In turn, other FSOs may forward credit card transactions to the credit card institution. Transactions for credit cards issued by the credit card institution may be processed by the credit card institution.

In the above example, the fee charged for each transaction, also called the merchant transaction price, is an example of a processing parameter for an FSO system in a credit card institution. One embodiment of an FSO system database in a credit card institution may include a merchant transaction pricing PCD table. The merchant transaction pricing PCD table may include one or more merchant transaction pricing values. Each merchant transaction pricing value may be associated with one unique key value in the table. The key values in the PCD table may be constructed using a key definition. Each processing parameter in the FSO system, and thus each processing parameter PCD table, may be associated with a key definition. In one embodiment, the FSO system database may include a key definition table for storing key definitions in the FSO system.

A key definition may include one or more data elements from the data dictionary. As an example, the merchant transaction pricing parameter described above may have a key definition that includes one or more data elements. Examples of data elements that may be included as fields in the merchant transaction pricing parameter key definition include card issuer, card type, on us/not on us, and transaction type. A card issuer may be the brand of card, for example, VISA, MasterCard, Discovery, etc. Examples of card types may include, but are not limited to: "gold" and "platinum" cards issued by some card issuers. On us/not on us refers to whether the FSO processing the transaction also issued the credit card. "On us" may mean that the FSO did issue the card. "Not on us" may mean that another FSO issued the card, and thus the transaction may be forwarded to the other FSO for processing. The term "transaction type" may refer to the way the

transaction was entered; examples of transaction types may include, but are not limited to: manual, electronic, and telephone transactions. A manual credit card transaction may be a credit card transaction that is entered by hand and imprinted with a credit card imprint machine. An electronic transaction may be a credit card transaction where the magnetic strip on a credit card is read electronically. A telephone transaction may be a credit card transaction performed by telephone call.

In Figure 1, an embodiment of a financial service organization business transaction processing system 100 may include a computer operating system 110, a database 130 comprising one or more data sets 131 residing on a data storage system 120. System 100 includes memory (not shown) configured to store computer application programs 140 for execution on system 100, and a central processing unit (not shown) configured to execute instructions of computer application programs 140 residing on system 100. A user 170 of a financial service organization business transaction processing system 100 may interact with the system via an application program 140. In one embodiment, an Application program 140 may process data sets 131 included in database 130 using random event based data set processing 160. In another embodiment, an Application program 140 may process data sets 131 included in database 130 using schedule based data set processing 150.

Figure 2 illustrates one embodiment of a financial service organization business transaction processing system with Smart Trigger software. In this embodiment, a financial service organization business transaction processing system 200 may include a computer operating system 210, a database 230 comprising one or more data sets 280 residing on a data storage system 220. System 200 includes memory (not shown) configured to store computer application programs 240 for execution on system 200, and a central processing unit (not shown) configured to execute instructions of computer application programs 240 residing on system 200. A user 270 of a financial service organization business transaction processing system 200 may interact with the system via

an application program 240. In one embodiment, an Application program 240 may process data sets 280 included in database 230 using random event based data set processing 260. In another embodiment, an Application program 240 may process data sets 280 included in database 230 using schedule based data set processing 250. In
5 another embodiment, an Application program 240 may process data sets 280 included in database 230 using Smart Trigger based data set processing 270.

Figure 3a is a data flow diagram illustrating a Smart Trigger selection for an associated executable processing task according to one embodiment. The Smart Trigger
10 table according to one embodiment comprises a list of pointers to an account data set 310 which need further processing as identified by a specific activity number 330 using activity data 340 on a user specified scheduled date 320. The specific activity number 330 is used as a key to access the associated processing task number 350 which in turn is used as a key to access an executable processing task name 370.

15 Figure 3b is a diagram illustrating a Smart Trigger date dependent data set selection according to one embodiment. The scheduled date being equal to or greater than today's date is a trigger to process the specific account data set 310 linked to the specific processing task name 370 as explained in Figure 3a.

20 Figure 4a is a flowchart illustrating the configuration and runtime use of prior art data set processing method for an FSO computer system, according to one embodiment. In one embodiment, steps 400 and 405 may be performed in a business transaction processing program in the FSO system. The user of the FSO computer may START the data set processing method by initiating a business transaction. In step 400, a user of the
25 FSO computer system may set up a TRIGGER file to process FSO related data sets on a specific scheduled date. The user would further define a series of evaluation criteria for every record in the FSO data set to determine if FSO computer needs to execute one or more processing tasks. For example, one embodiment of a FSO related data set may be

an FSO account master file. An embodiment of a processing task may determine if the credit card expiration date falls within the next two months and if it does the processing task may further include extending the expiration date of the FSO issued credit card by one year. In step 401, the FSO data set processing software may be compiled, generated and downloaded. In step 402, the user may initiate the task to process the TRIGGER FILE. In step 403, the trigger file is processed sequentially. The first record in the file which may consist of a data set pointer and a scheduled date may be read. In step 404, a comparison may be made between the scheduled date and the current date to determine if the current record of the FSO data set needs further processing. As one embodiment, if the current date is greater than or equal to the scheduled date then the current record may need further processing. In step 405, if the current date is less than the scheduled date then the next sequential record in the FSO data set may be read and the step 404 may be executed again.

Figure 4b is a continuation of flowchart in Figure 4a. Step 410 is executed after step 404 in Figure 4a. In step 410, the data set pointer may be used to access the entire record in the FSO data set. In step 413, the data contained in the FSO record may be evaluated to determine if criteria for executing one or more processing tasks have been met. If the one or more evaluation criteria are matched then step 414 would be executed next. If the one or more evaluation criteria are not matched then step 412 would be executed next. As one embodiment of the criteria for executing a processing task to extend the credit card expiration date, an evaluation may be made to determine if the current credit card expiration date falls within the next two months. In step 414, if the data in the FSO record met the criteria then the processing task may be executed. As one embodiment, the processing task may extend the expiration date of the FSO issued credit card by one year for the current FSO record. In step 416, an evaluation may be made if there are any additional processing tasks for the current FSO record. If it is determined that there may be additional processing tasks then step 415 would be executed next. If it is determined that there may be no additional processing tasks then step 420 would be

executed next. In step 412, since it has been determined that one or more evaluation criteria were not matched, a further evaluation may be made to determine if additional processing tasks remain to be processed for the current FSO record. If no additional processing tasks remain to be processed for the current FSO record then the program control would go to step 420 in Figure 4c. If additional processing tasks remain to be processed for the current FSO record then step 411 would be executed next. In step 411, the next processing task for the current sequential record in the FSO data set may be accessed and program control may be transferred to step 413 in Figure 4b. In step 415, the next processing task for the current sequential record in the FSO data set may be accessed and program control may be transferred to step 413 in Figure 4b.

Figure 4c is a continuation of flowchart in Figure 4b. In step 420, the scheduled date may be reset to indicate that all processing tasks for the current FSO record may have been completed. Program control may then be transferred to step 422. In step 422, an evaluation may be made if there are any additional records of the current FSO data set. If there are additional records to be processed then program control may be transferred to step 421. If there are no additional records to be processed then program control may be transferred to the END step. In step 421, the next sequential record for the current FSO data set may be accessed and program control may be transferred to step 404 in Figure 4a.

Figure 5 is a flowchart illustrating the configuration and runtime use of a Smart Trigger based processing method for an FSO computer system, according to one embodiment. In one embodiment, steps 500 and 509 may be performed in a business transaction processing program in the FSO system. A user of the FSO computer may START the data set processing method by initiating a business transaction. In step 500, the user of the FSO computer system may configure a Smart Trigger by defining an identifier for an FSO data set, a scheduled date and an identifier for an associated processing task. In another embodiment, the user may also define a data field with the

associated processing task identifier. As one embodiment, the user may set up a Smart Trigger table for processing one or more accounts of an FSO data set on a particular scheduled date by assigning an associated task to be executed when certain pre-defined condition such as current date is equal to or greater than the scheduled date has been met.

5 In step 501, the user of the FSO computer system may schedule a task to be executed at a user defined period to process the Smart Trigger table. In step 502, an evaluation may be made if certain predefined condition such as execution date and time for processing Smart Trigger task is equal to or greater than the current date and time has been met. If the scheduled conditions are met then the Smart Trigger table may be ready for

10 processing in step 503. If the scheduled conditions have not been met then step 502 will loop on itself. In step 503, the processing of the Smart Trigger table may be initiated by reading the first row of the Smart Trigger table. In step 504, an evaluation may be made if certain predefined condition such as scheduled date field obtained by reading the data from the first row in the Smart Trigger table is equal to or less than the current date has

15 been met. If the predefined date condition has been met then program control is passed on to step 506. If the predefined date condition has not been met then program control is passed on to step 505. In step 505, the current row index for the Smart Trigger is incremented by one. After reading the data for the new row, program control is passed on to step 504. In step 506, if the data contained in a row of the Smart Trigger table met the

20 predefined date conditions then the associated processing task may be executed. As one embodiment, the processing task may extend the expiration date of the FSO issued credit card by one year for the current FSO record. In step 507, the scheduled date field in the Smart Trigger table may be reset to indicate that the processing task for the current row of the Smart Trigger table may have been completed. In step 508, an evaluation may be

25 made if there are any additional rows of the Smart Trigger table. If there are additional rows to be processed then program control may be transferred to step 509. If there are no additional rows to be processed then program control may be transferred to the END step. In step 509, the next row for the Smart Trigger table may be accessed and program control may be transferred to step 504 in Figure 5.

Figure 6 is a data flow diagram illustrating sequential processing of a FSO data set in prior art, according to one embodiment. A data set may consist of one or more data records. A data set record pointer 600, which may vary from P1 through PN, may be used
5 as a pointer to the physical storage location of the data set record. The data in the data set record may be stored in data column #1 610 through data column #N 620. For each data set record, one or more evaluations may have to be made for every processing task T1 through TN 630. The data set processing may be initiated by reading data set pointer P1. Data set record 610 and 620 associated with P1 may be evaluated to determine if
10 processing task T1, T2, T3 and others through TN need to be executed. In one embodiment, 650 may store the current processing task and 660 may store the next processing task. Once all processing tasks T1 through TN associated with the first data set pointer P1 have been completed then the same process can be repeated for data set pointer P2. The data set sequential processing may continue by reading data set pointer
15 P3 through PN and evaluating and executing processing tasks T1 through TN for each data set pointer.

Figure 7 is a data flow diagram illustrating selective task processing of a FSO data set using Smart Trigger, according to one embodiment. A data set may consist of one or
20 more data records. A data set record pointer 700, which may vary from P1 through PN, may be used as a pointer to the physical storage location of the data set record (not shown). A selective task processing method using Smart Trigger may build a list of associated data set pointers for a specific processing task T1 710. In one embodiment, data set records with data set record P2, P5 and PN have been specifically identified to
25 execute processing task T1. Similar lists may be built for processing tasks T2 through TN. Data contained in the data records may be accessed by tasks T1 through TN using associated data set pointers 700.

Figure 8 is a block diagram illustrating a scheduled task processing entries in a

Smart Trigger table, according to one embodiment. The user of the FSO computer system may schedule a task to be executed at a user-defined period to process the Smart Trigger table. An evaluation may be made if certain predefined condition such as execution date and time for processing Smart Trigger task is equal to or greater than the current date and
5 time has been met. If the scheduled conditions are met then the Smart Trigger table may be ready for processing by reading the smart trigger table 810. If the scheduled date included in the smart trigger table record meets the trigger conditions 850 then the associated task may be processed 870. The associated processing task 870 may use specified data set 860 to generate results 880. On completion of the processing of the
10 associated task 840, the scheduled date may be updated in the smart trigger table 830 to indicate its completion.

Various embodiments further include receiving or storing instructions and/or data implemented in accordance with the foregoing description upon a carrier medium.
15 Suitable carrier media include memory media or storage media such as magnetic or optical media, e.g., disk or CD-ROM, as well as signals such as electrical, electromagnetic, or digital signals, conveyed via a communication medium such as networks and/or a wireless link.

20 Although the system and method of the present invention have been described in connection with several embodiments, the invention is not intended to be limited to the specific forms set forth herein, but on the contrary, it is intended to cover such alternatives, modifications, and equivalents as can be reasonably included within the spirit and scope of the invention as defined by the appended claims.

25